IGVC Auto-Nav Challengeのための全方位画像を用いた白線認識およびNavigation用ROS componentsの開発

著者	河野 辰哉
出版者	法政大学大学院理工学研究科
雑誌名	法政大学大学院紀要．理工学・工学研究科編
巻	60
ページ	1-6
発行年	2019-03-31
URL	http://doi.org/10.15002/00022067

IGVC Auto-Nav Challenge のための 全方位画像を用いた白線認識および Navigation 用 ROS components の開発

Development of ROS navigation components using omni-directional images-based lane recognition for IGVC Auto-Nav challenge environment

河野辰哉
Tatsuya Kawano
指導教員　小林一行 教授

法政大学大学院理工学研究科システム理工学専攻修士課程

This paper describes the development of Robot Operating System (ROS) component for lane recognition and navigation of IGVC Auto - Nav Challenge. To achieve a robust and stable navigation, we propose new lane and obstacle recognition algorithm based on omnidirectional camera. Employing fast NCC based piecewise template matching technique enables robust lane detection regardless of surrounding brightness changes and various lane shape including sharp curves. To recognize surrounding obstacles only vision sensor without LiDAR, we employ YOLOv3-tiny based deep learning algorithm which can avoid obstacle collision. Since ROS has capability of environmental simulation by using Gazebo, rapid prototyping is achieved by applying both simulations and actual experiments which can significantly reduce development period. The effectiveness of the newly proposed algorithm was confirmed by both offline simulation and actual outdoor experiment.

Key Words: *Mobile robot, Lane detection, Omni-directional camera, Robot Operating System, YOLO*

１．はじめに

これまでの移動ロボットは工場内などの整備された環境下で主に利用されてきた. 近年は, センサの高性能化や人工知能技術の発展による環境認識技術の向上を背景に, 自動走行技術を応用した人の道案内や巡回警備など，実環境下で機能する移動ロボットの研究開発が盛んに行われている.

筆者の所属する自律ロボット実験室は，屋外環境下を想定した自律移動ロボットを研究対象とし，毎年アメリカで開催される IGVC (Intelligent Ground Vehicle Competition) [1]に参加している．IGVC の詳細については次項で紹介する．本研究では，IGVC のメイン競技である Auto-Nav Challenge のための白線認識とナビゲーション用 ROS Component の開発および検証を行なう.

２．IGVC の概要

Fig.1 に IGVC のようすを示す．IGVC は，屋外地上自律走行車に関する国際的な競技大会であり，国際自律走行車協会(AUVSI：Association for Unmanned Vehicle Systems International) の主催により，毎年アメリカで開催されている．出場する移動ロボットは無線などにより遠隔で操作を行わず，ロボットに搭載されたセンサの情報を用いて環境を認識し，目的に合わせた自律走行を行なう．大会は Auto-Nav Challenge（ナビゲーション部門），Design Competition（デザインプレゼンテーション部門），IOP Challenge（通信部門）の 3 つの競技部門から成り，その総合得点で順位を競う.

Fig.1 IGVC

Fig.2 に，Auto-Nav Challenge の競技環境を示す．コースはゴルフコースのような芝生の場所にあり，白線の間

を走行するエリアと，大会運営側が設定した GPS 座標（ウェイポイント）を正確に通過するウェイポイントナビゲーション[2]エリアの 2 つで構成される．出場チームはこのコースの走行距離と完走タイムを競う．ウェイポイントは直径 2m の円でコース上に表わされ，移動ロボットはその円内の一部を通過すればよい．またコース内には複数の静止障害物が配置されており，それらを避けながら走行する必要がある．

Fig.2 Auto-Nav Challenge course

３．ROS（Robot Operating System）について

本研究では，ソフトウェア開発環境に ROS(Robot Operating System)の Kinetic バージョンを採用した．ROS は，Open Source Robotics Foundation (OSRF）が管理する世界で最も利用されるロボットフレームワークである．ROS の特徴として，Fig.3 に示すようなロボット開発用のライブラリやツールがオープンソースで提供されており，それらが多くのロボットやセンサに対応していることなどが挙げられる．

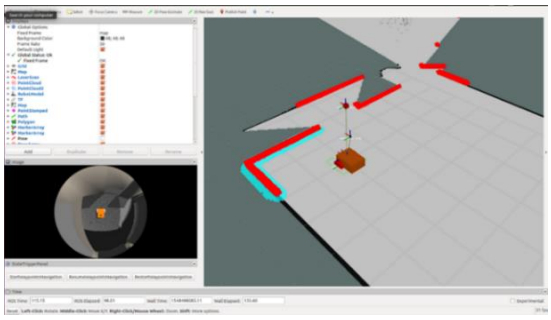


Fig.3 ROS visualization tool (Rviz)

Fig.4 に，ROS のメッセージ通信の概要を示す．ROS で実行するプログラムは Node という単位で扱われ，Master は Node 間の通信を管理するために，メッセージの登録や検索を行なう．Publisher は Topic にデータを書き込み配信し，Subscriber がその Topic を購読することでデータをやり取りする．また ROS では，扱うデータの種類によってメッセージのデータ構造が定められており，その構造にあわせてメッセージを作成する必要がある．ROS はこのように分散システムで構成されている点や，メッセージのデータ構造が定められている点などから，部分的なソフトウェアエラーがシステム全体の運用に影響を与えづらく，各プログラムの再利用性が高いなどのメリットがある．

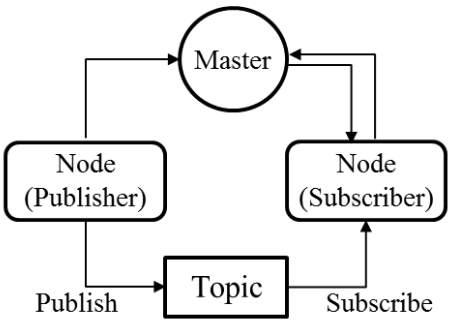


Fig.4 ROS data communication system

４．提案手法のための移動ロボットの構成

（１）移動ロボットの概要

Fig.5 に，Auto-Nav Challenge に出場するために開発した移動ロボットのシステム構成を示す．全方位カメラは大会ルールの上限値である地面から 180cm の高さにカメラレンズと地面が平行になるように設置する．これにより移動ロボットの周囲 360 度の画像を一度に取得することができる．画像処理は，GPU 演算に特化した組み込み PC である JetsonTX2 で行ない，その処理結果のみがノート PC に送信される．その他のセンサデータの処理や移動ロボットの制御はノート PC で行なう．ROS は，ノート PC 側でマスターを立ち上げ，そのマスターを JetsonTX2 でも共有することで，二台のコンピュータ間で ROS メッセージのやり取りを行なう．Table.1 に各コンピュータのスペックを記載する．

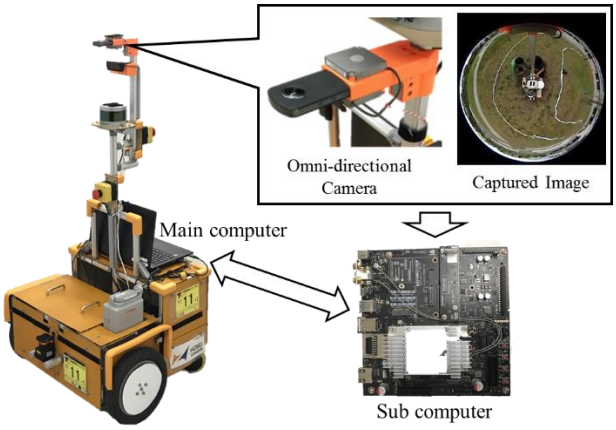


Fig.5 Robot configuration

Table.1 Computer Specs

	Main Computer (FRONTIER FRNXW610/c)	**Sub Computer (JetsonTX2)**
OS	Ubuntu16.04	Ubuntu16.04
CPU	Intel Core i7 2.5GHz	HMP Dual Denver 2/2 MB L2 + Quad ARM® A57/2 MB L2
Memory	16GB	8GB
GPU	None	Pascal CUDA core 256

５．仮定と問題の記述

（１）仮定の記述

Auto-Nav Challenge のための，白線認識およびナビゲーション用システムを提案する上での仮定を以下に示す．

A）走行環境は Auto-Nav Challenge の環境を想定する．

本仮定は，IGVC の Auto-Nav Challenge のルールに基づいて設ける．本研究に関連する Auto-Nav Challenge のルールを以下に示す．

R1）白線は緑色の芝生上に引かれている．

R2）動的障害物はない．

R3）白線自体の幅は約 3 inch (7.6cm)

R4）コース幅は約 10 feet (300cm)

（２）問題の記述

Auto-Nav Challenge のための，白線認識およびナビゲーション用システムを開発する上での問題を以下に示す．

P1）どのように画像から白線と障害物を認識するのか．

P2）どのように白線からの逸脱を防止し，障害物を回避しながら走行するのか．

P3）どのように開発効率を上げるのか．

６．提案するシステム

Fig.6 に，提案システムのアルゴリズムを示す．最初に全方位カメラから取得した画像を地平面変換し，テンプレートマッチングによって白線を抽出する．つぎに白線抽出画像をパノラマ画像に変換し，その画像から白線の位置を特定する．障害物認識は，機械学習アルゴリズムの YOLOv3[3]を軽量化した YOLOv3-tiny により行ない，その結果から障害物の位置を特定する．最後に，それぞれのデータからコストマップを生成することで，白線を逸脱せずに障害物を回避する経路生成を行なう．次節より一連の処理の詳細について述べる．

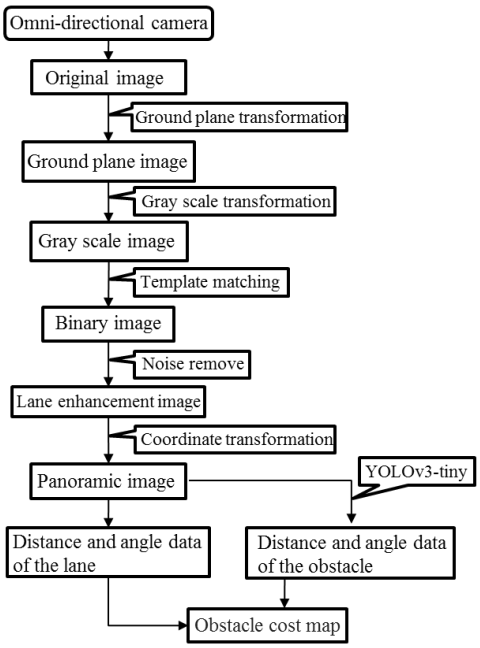
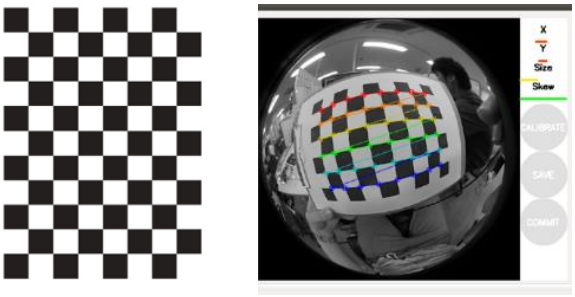


Fig.6 Development lane and obstacle recognition algorithm

（１）白線認識アルゴリズム

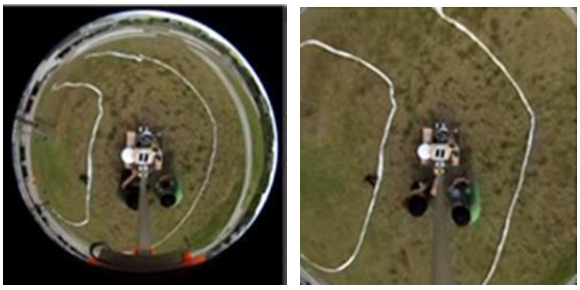
a）全方位画像の地平面変換

本研究で用いる全方位カメラは魚眼レンズであり，取得した元画像が歪みをもつため，白線を正確に検出するために歪み補正を行なう．そこで Fig.7 に示すように，チェッカーボードと ROS のパッケージ camera_calibration[4]を使用し，全方位カメラのパラメータを求める．そして，そのパラメータと OpenCV の歪み補正関数により地平面画像に変換する．Fig.8 に変換結果を示す．



(a) Checker Board (b) Calculate camera parameter

Fig.7 Camera calibration



(a) Original image (b) Ground plane image

Fig.8 Result of distortion correction

b）白線候補点の抽出

白線抽出はテンプレートマッチングにより行なう．アルゴリズムには NCC (Normalized Cross Correlation) [5]を採用した．NCC は照明変化に強いという特徴を持ち，屋外環境での天候変化の影響を減らすことができる．テンプレートマッチングにかけるグレースケール画像には，RGB 画像の，白線部分がより強調される B(Blue)成分の画像を使用した．Fig.9 に，使用したテンプレート画像の概要を示す．120×120 [pixel] のグレースケール画像に幅 3inch の白線はおよそ 4 pixel でうつる．そこでその白線やカーブを検出できる，縦横 2 種類のテンプレート画像を作成した．Fig.10 に，テンプレートマッチングにより白線類似度の高さを表わした三次元グラフを示す．この類似度が一定以上の場合，その場所を白線候補点と判断し二値化画像に残す．Table.2 に白線抽出の結果を示す．直線以外のカーブや輝度の低い条件下においても白線を抽出できていることが確認できる．

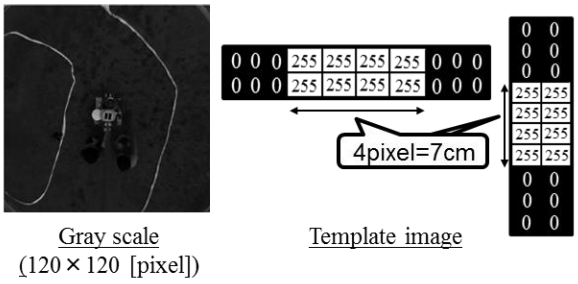


Fig.9 Template images

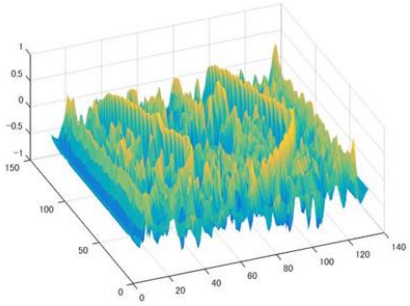


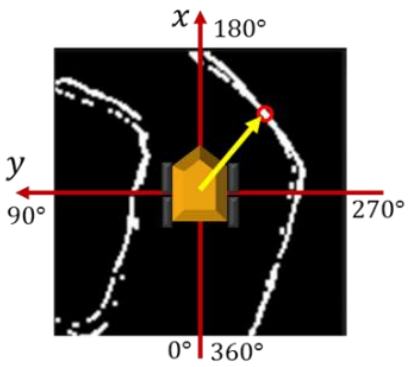
Fig.10 Potential Field

Table.2 Results of template matching

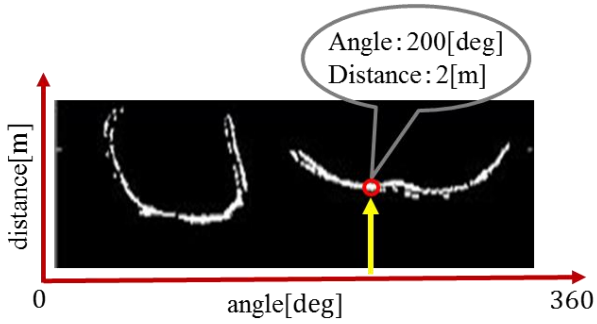
	Straight	Curve	Dark
RGB			
Lane Extract			

c）白線位置の距離・角度データの取得

白線抽出画像から，移動ロボットを原点とした白線までの距離と角度情報を取得する．距離と角度を求めやすくするために，Fig.11(a)の白線抽出画像を Fig.11(b)のようなパノラマ画像に座標変換する．これにより，横軸が白線までの角度，縦軸が白線までの距離を表わした画像となり，容易に白線の位置情報を求めることができる．



(a) Lane extraction image



(b) Panoramic image

Fig.11 Coordinate transformation

（2）ナビゲーションアルゴリズム

a）ナビゲーションの概要

認識した白線を逸脱せず障害物回避をしながら走行する手法について述べる．自律走行時の障害物回避や経路計画は，ROS のナビゲーションパッケージ move_base[6]により行なう．move_base は，LiDAR などの二次元距離センサに用いる LaserScan 型メッセージを購読することでコストマップを生成し，移動ロボットがそのコストマップ内に侵入しないような経路計画を行なう．コストマップは，ロボットが移動する二次元空間をグリッドに分割し，各グリッドで障害物がある確率を数値化した占有格子地図である．Fig.12 に，実際に LiDAR で通路中央の障害物と壁をコストマップ化したようすを示す．コストマップの黄色の部分が実際に障害物の位置を示し，水色の部分は移動ロボットが侵入すると障害物と衝突する領域を示している．走行経路は，コストマップに侵入しないような方向に生成されている．Fig.13 に本手法の概要を示す．全方位カメラで認識した白線と障害物の情報からコストマップを生成し，白線を逸脱せず障害物回避が可能なナビゲーションシステムを提案する．

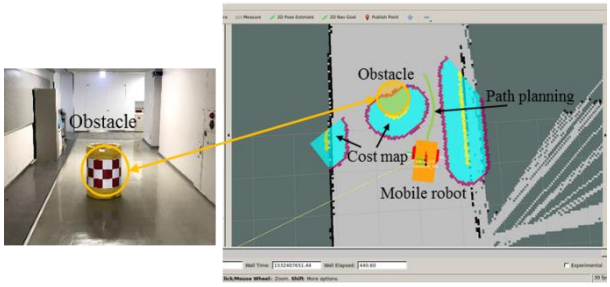


Fig.12 Costmap and path planning

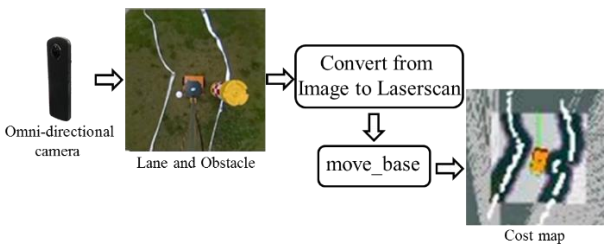


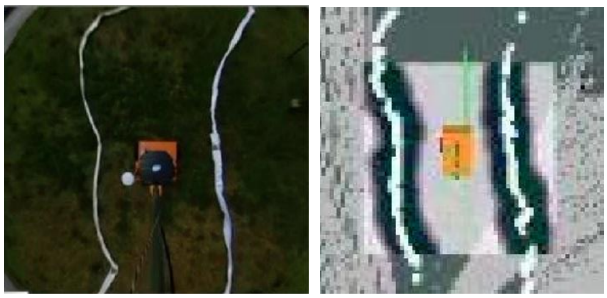
Fig.13 System overview

b) 白線のコストマップ化

白線をコストマップ化する手法を述べる．白線をコストマップ化するには，まず Fig.11(b)に示したパノラマ画像の Image 型メッセージから，LaserScan 型メッセージを作成する必要がある．Fig.14 に示すように，LaserScan 型メッセージは測定角度範囲(angle_min,max)，角度分解能(angle_increment)，スキャンタイム(scan_time)，測定距離範囲(range_max,min)，測定結果(ranges, intensities)などで構成される．パノラマ画像から求めた白線の位置データは ranges に格納する．そして作成したメッセージを move_base に配信し，白線をコストマップ化する．Fig.15 に変換結果を示す．Fig.15(b)の白い線が白線の LaserScan 型データを示し，白い線の周りがコストマップ化されていることが確認できる．

```
std_msgs/Header header
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
float32 range_min
float32 range_max
float32[] ranges
float32[] intensities
```

Fig.14 Message type definition of LaserScan



(a) RGB image (b) LaserScan and costmap

Fig.15 Convert from image to costmap

c) 障害物のコストマップ化

Auto-Nav Challenge のコース上には，障害物としてクッションドラムが配置されている．従来は距離センサによってこの障害物を認識していたが，本手法では全方位カメラのパノラマ画像から障害物認識を行なう．障害物の認識アルゴリズムには，機械学習フレームワークの Darknet に実装される YOLOv3-tiny を採用した．YOLOv3-tiny は YOLOv3 の計算処理を軽量化したモデルであり，JetsonTX2 での処理速度は約 20[Hz]である．障害物の学習は，学習データ 100 枚，エポック数 10000 で行なった．

Fig.16 に学習結果を用いて障害物を認識した結果を示す．認識した障害物は四角の Bounding Box で囲われ，画像中での障害物座標がわかる．この座標情報を用いることで，白線認識手法と同様に障害物までの距離と角度を算出し，LaserScan 型メッセージを作成することでコストマップを生成する．提案手法による検証を Fig.17(a)の環境で行った．Fig.17(b)の青と赤で示された部分が LiDAR の LaserScan データであり，白の部分が提案手法による LaserScan データである．障害物の位置でそれぞれのデータが重なり合っていることから，Bounding box から算出した距離が有効であるとわかる．また Fig.17(c)には，その障害物をコストマップにした結果を示す．

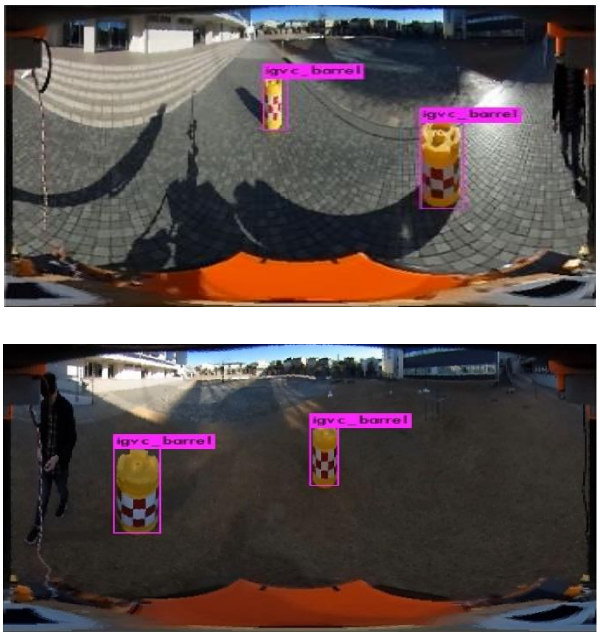
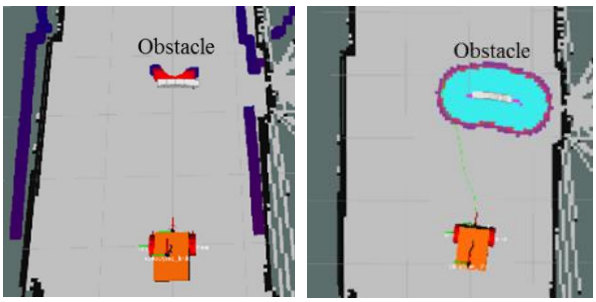


Fig.16 Results of obstacle detection by using Yolov3-tiny

(a) Mobile robot and obstacle



(b) LaserScan data (c) Costmap

Fig.17 Convert from bounding box to LaserScan and costmap

７． 実験

（１）シミュレーション実験

移動ロボットの研究開発では，開発したシステムの検証やパラメータ調整が重要である．そこで，Gazebo シミュレータで競技環境を構築し，Auto-Nav Challenge での走行を再現することで開発の効率化を図った．また本研究で用いる ROS の Kinetic バージョンから魚眼レンズカメラのモデルが追加され，全方位カメラを用いたシミュレーションも可能となった．ロボットモデルの作成には，ロボットの構造を記述するための URDF(Unified Robot Description Format)を用いて行なう．

a) Gazebo による Auto-Nav Challenge 環境の構築

Fig.18 に Gazebo で作成した Auto-Nav Challenge の競技環境を示す．地面の芝生と白線は画像編集ソフトで作成した画像を使用し，コース上には静的障害物のクッションドラムを配置した．

Fig.18 Auto-Nav Challenge filed in Gazebo

b) シミュレーション

Fig.19 に Gazebo による走行シミュレーションのようすを示す．開発した一連のシステムを即座に検証できるようになったため開発効率が向上し，シミュレーション実験の有効性を確認できた．

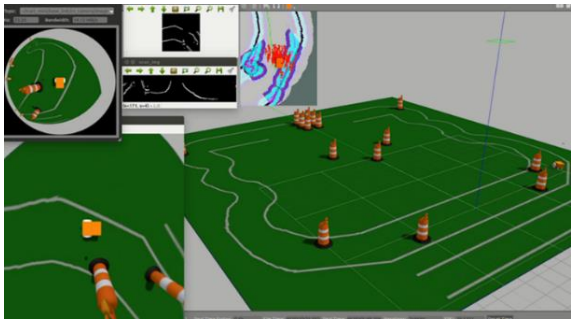


Fig.19 Simulation experiment

（２）実機実験

提案システムの有効性を確認するために，屋外での実機実験を行った．Fig.20 に実験環境，Fig.21 にナビゲーション中に白線と障害物を認識した結果の一部を示す．コースの形状や障害物の有無に影響せず，安定して走行できることを確認した．

Fig.20 Experimental enviroment

Fig.21 Experimental result in real environment

８． 結論

本研究では，IGVC Auto-Nav Challenge 出場のために，全方位カメラを用いた白線認識とナビゲーション用 ROS コンポーネントの開発を行なった．Auto-Nav Challenge の競技環境を再現した Gazebo によるシミュレーション実験と，屋外での実機実験を行なった．その結果，あらゆる条件下においても，白線と障害物を認識しながら安定してナビゲーションできることを確認し，提案システムの有効性が証明された．

９． 今後の展望

今後は，白線認識に対しても機械学習を導入し，白線認識のロバスト性のさらに高めていきたい．また障害物認識システムについては，現地にある実際の障害物データが少なかったため，今回は学内で用意したクッションドラムで代用し，学習と実験を行なった．そのため，本システムの本格的な実装は次回の IGVC 以降になる．

参考文献

1) Intelligent Ground Vehicle Competition: http://www.igvc.org/　2018/02/11 アクセス

2) 御園佑介:複素拡張カルマンフィルタを用いたウェイポイントナビゲーションの構築，2009 年度法政大学大学院工学研究科システム工学専攻修士論文，2010

3) Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: YOLOv3: An Incremental Improvement, arXiv:1804.02767,2018

4) ROS Wiki camera_calibration: http://wiki.ros.org/camera_calibration

5) J.P. Lewis: Fast Template Matching, Vision Interface 95, Canadian Image Processing and Pattern Recognition Society, Quebec City, Canada, May 15-19, p. 120-123, 1995

6) ROS Wiki move_base: http://wiki.ros.org/move_base